

SERVICE BROWSER PROCESS AND RELATED SYSTEM**Field of the invention**

The present invention relates to a service browser process and the corresponding system.

A service browser process is a process for controlling navigation between a plurality of services and/or channels in a digital Integrated Receiver Decoder (IRD) that provides links to other services within a radio-broadcasting or TV environment, or across TV networks, and allows the viewer to tune to those services.

Services are regular TV or radio services, or any application or communication data channel as for instance indicated in the DVB service descriptor of the MPEG-2 standard.

A service consists of one or more components or tracks transmitting video, audio or application data.

Background of the invention

The process of changing from one TV channel to another is often called « channel zapping » or « channel surfing ». This particular type of browser application will be referred to hereafter indifferently as « Surfer application » or « Surfer ».

A Surfer logically can be seen as a gate to a selection of services or list of services.

In other words, Surfer is used as a general term describing a broadcasted application that is designed to handle the presentation of a list of services and will react to particular events like remote control 5 key messages, but is not related to a specific application. It is also already observed that a Surfer is one level above the other applications hereafter referred to as normal applications or EPG applications as defined in the DVB standard.

10 The invention is more particularly but not exclusively related to a process and a system implemented within the digital interactive TV environment.

In digital TV Broadcast environment any 15 interactive decoder is decoding video, audio and data streams, regardless of the fact that it is connected to a cable, satellite or terrestrial network.

For instance, [1] ETS 300 468 Specification for Service Information (SI) in Digital Video 20 Broadcasting (DVB) systems, and [2] ISO/IEC DIS 13818 -1, Generic coding of moving pictures and associated audio information; Part 1 Systems, describe ways to include multiplex and retrieve single data streams into different broadcast environments.

25 More generally it is observed that applications are transmitted in data streams and processed in the decoder's memory.

The Surfer of the prior art is not broadcasted and consists of a built-in application, called Banner,

stored in the decoder in order to allow a user to navigate or surf through all available services.

This built-in software presents all services to the viewer and enables the channel changing process.

5 In other words the Banner handles all channel-changing processes inside the decoder.

More particularly apart from internal set-up menus, the Banners of known IRD incorporates an embedded way presenting one or more lists of services

10 to the TV viewer, for instance one TV list and one radio list. These lists and all navigation functions within these lists, are controlled by the manufacturer's navigation software.

For such control of the navigation software, the
15 relevant network operator has previously provided the needed instruction to the manufacturers of the IRD.

This is for instance the case of PayTV networks, renting their decoder platform.

However, the built-in Banner involves two main
20 disadvantages in the channel navigation concept, i.e. the navigation interface looks different on each type of decoder, and competing broadcasters or service providers that share the decoder platform do not have any control on promoting their own channels or
25 services.

Summary of the present invention

The present invention aims at solving these drawbacks and disadvantages.

For this it is a main object of the present invention to provide a process and a system which render possible to download within the decoder the Surfer corresponding to the selected application, and to display such Surfer on a real-time basis, therefore providing to the user the required look and feel and services corresponding to the relevant provider, while maintaining reception of the selected service and authorising reception of other services coming from other sources (transponders) different from the one of the Surfer.

It is another object of the invention to provide navigation control functions to a broadcasted Surfer application, to allow running the same browser on different IRDs from different manufacturers.

This guarantees the same user interface and behaviour across various decoders, while embedded browsers come with their individual look and feel different from the built-in software.

It is another object of the invention to provide an incorporated and cost saving process and system for perfect surfing.

Specifically, the invention proposes a service browser process for controlling navigation events between a plurality of services and/or channels of a digital interactive Radio-Broadcasting system including at least one digital interactive decoder,

said system broadcasting applications to be received by said decoder, wherein the process proposes said services to a user of said decoder and enables the navigation to other services or channels through control means activated by said user, characterised in that

the applications being categorised into at least two types of applications including a first type termed Surfer application designed for controlling said navigation and having knowledge of said services,

the process comprises the steps of :

- (i) identifying Surfer applications from other types of applications,
- (ii) selecting a particular Surfer application,
- (iii) downloading such selected Surfer application within a dedicated part of the decoder memory, called Surfer Cache, and
- (iv) executing said selected Surfer application from said Surfer Cache, whereby the decoder is under control of said Surfer application.

When a Surfer is started, it allows the user to tune to another channel while still remaining active.

In an advantageous embodiment, the decoder comprising a built-in application for presenting the services, termed the built-in Banner, and once a Surfer application is stored within said Surfer Cache, for any navigation event, the process comprises the steps of :

- checking if said navigation event has to be forwarded to the built-in Banner or to the Surfer application, and

5 - in case the decoder is controlled by said Surfer application, routing said navigation event to the Surfer application while the built-in Banner is disabled.

By navigation event one should understand any event of navigation between services such as for 10 instance channel up or channel down.

Advantageously, the Surfer application is stopped when an application different from a Surfer application, termed normal application, is displayed, and the Surfer application is re-launched from its 15 Surfer Cache when said normal application is finished or a navigational event occurs.

Equally advantageously a plurality of Surfer applications being possible, it comprises the step of :

20 - presenting an interface using a list of services that allows the user to select one particular Surfer application among said list and to come back to said list after selection, if wanted.

In a preferred embodiment, the process is 25 implemented in a DVB environment, the Surfer applications being signalled in Bouquet Association Tables (BAT).

A bouquet is a broadcasted service list.

Advantageously, and in order to reduce switching time between several bouquets (i.e. service lists) containing Surfer applications, the process includes the step of downloading a plurality of Surfer applications, within corresponding Surfer Caches. 5 Several Surfers can therefore be prefetched into these different caching areas inside the decoder's memory, and are launched when or after the viewer has requested a certain bouquet or service list.

10 Advantageously, a unique identifier is provided to attach a Surfer application to a bouquet or service list in order to retrieve the proper Surfer application on viewer's bouquet selection.

15 The system is then able to tune directly to one of the bouquet's service and bring up the Surfer from its Cache.

This allows a quicker service as it is not necessary to download the relevant Surfer when needed.

20 The invention also provides a system for implementing the process as described above.

More particularly, the invention also provides a digital Interactive Radio-Broadcasting system for controlling navigation events between a plurality of 25 services and/or channels, including at least one digital interactive decoder, said system broadcasting applications to be received by said decoder, wherein the system proposes said services to a user of said decoder and enables the navigation to other services

or channels through control means activated by said user, characterised in that

the applications being categorised into at least two types of applications including a first type termed

5 Surfer application designed for controlling said navigation and having knowledge of said services,
the decoder comprises:

(i) identifying means for identifying Surfer applications from other types of applications,

10 (ii) selecting means for selecting a particular Surfer application,

(iii) downloading means for downloading such selected Surfer application within a dedicated part of the decoder memory, called Surfer Cache, and

15 (iv) calculating means for executing said selected Surfer application from said Surfer Cache, whereby the decoder is under control of said Surfer application.

Advantageously, the decoder comprising a built-in application for presenting the services, termed the built-in Banner, once a Surfer application is stored within said Surfer Cache, for any navigation event, the system comprises :

- checking means for checking if said navigation event has to be forwarded to the built-in Banner or to the Surfer application,

- routing means arranged for routing said navigation event to the Surfer application in case

the decoder is controlled by said Surfer application,
and

- disabling means for disabling simultaneously
the built-in Banner.

5 Also advantageously, the system comprises stopping
means for stopping the Surfer application when a
normal application is displayed, and re-launching
means for re-launching said Surfer application from
the Surfer Cache when said normal application is
10 finished.

In an advantageous embodiment, a plurality of
Surfer applications being possible, it comprises
means for presenting an interface using a list of
Surfers that allows the user to select one particular
15 Surfer application among said list and to come back
to said list after selection, if wanted.

Preferably the system is implemented in a DVB
environment, the Surfer applications being signalled
in Bouquet Association Tables (BAT).

20 Advantageously, there is a plurality of Surfer
Caches.

More categories can also be selected but what is
important is to be able to identify Surfer
applications.

25 More generally, the Surfer applications have or
may have the following properties:

- They must be stored in the decoder to control
channel navigation.

- A Surfer application is always running if there is no other normal application requesting to be executed.

5 - Surfer applications are controlling all navigation events.

- Surfer applications have knowledge on services, service order and service properties and Surfer applications are downloaded into the decoder each time the viewer selects a particular Surfer.

10 In another embodiment, such downloading is not necessary at each time, as a plurality of Surfers may be simultaneously stored in the decoder and retrieved at demand.

15 For this, Surfer controlled navigation requires an additional recognition process, physical memory for temporary caching the Surfers, and a built-in Surfer selection layer in the decoder's user interface.

- The recognition process is a software component in the system that is detecting a Surfer application.

20 - This process has knowledge about normal and Surfer applications.

25 According to the embodiment of the invention more particularly described here and while normal applications are directly loaded and executed into decoder's memory, a Surfer application will first be stored into the dedicated part of the decoder memory (Surfer Cache) and after it will be loaded and executed from its Cache.

Brief description of the drawings

The present invention will be better understood from reading the following description of particular embodiments given by way of non limiting examples, 5 with reference to the accompanying drawings, wherein :

Figure 1 shows a schematic diagram figuring the a TV broadcasting system to be used with the invention.

Figure 1bis is a more detailed block diagrams of 10 an Integrated Receiver Decoder (IRD) used with the invention.

Figure 2 shows a schematic view of the inside memory of a decoder of the prior art, with two examples of remote control key events accessing the 15 system for navigation purpose

Figure 3 shows a schematic view of the inside memory of a decoder according to one embodiment of the invention, with an example of a simplified flowchart navigations events accessing the system.

20 Figure 4 is a flowchart highlighting the fact that a service is selected either using a Banner or a Surfer, according to an embodiment of the invention.

Figure 5 is a functional diagram illustrating the paths a user/viewer could choose according to an 25 embodiment of the invention.

Figure 6 shows a flowchart figuring an example of Surfer implementation according to the invention.

Figure 7 is a flowchart giving an example of interactions between system, Surfer and normal applications according to the invention.

5 DETAILED DESCRIPTION

Figure 1 shows a digital Interactive Radio-broadcasting system 1 comprising an interactive decoder 2 to be connected to a TV set 3, receiving 10 audio signals A, video signals V and data streams signals D from broadcasting networks, i.e. the cable Network C, the terrestrial Network T, and/or the satellite network S.

The decoder 2 comprises (see figure 1bis), in a manner known per se, a demultiplexer 4 connected to an Audio/Video Decoding process circuit 5, a CPU 6 connected to a modem 7, a ROM 8 for storing the operating system of the decoder, such as the Open TV operating system, and a read/write memory 9, such as 20 a RAM for storing the broadcasted applications.

Figure 2 shows a simplified view of the inside memory 9 of a decoder of the prior art. The memory 9, typically a RAM of 2 Mbytes or more, includes a first part 10 storing some system applications of the 25 decoder, and a built-in Banner 11, or built-in navigator software, developed by the decoder manufacturer. The built-in Banner allows access to services.

In the DVB environment described in ETS 300 468 [1] incorporated herein by reference, the list of services is referred to as a « bouquet ».

As already mentioned a bouquet is a service list 5 which is broadcasted within a DVB Bouquet Association Table-BAT.

The memory 9 further includes a part 12 for storing a normal broadcasted application.

In the remaining part of the text it will also be 10 sometimes referred to elements mentioned in the MPEG-2 Standard, (system ISO/IEC 13818-1 of Nov. 1994 [3]) also incorporated herein by reference.

Figure 2 shows in 14, two examples of navigational events activated by a remote control device 15 (see 15 figure 1) of the prior art.

By pressing remote control key **P+** (key 16) or (key 20 [53], the user activates navigational events 18 « next channel » or 19 « go to channel 53 » which are forwarded to the built-in Banner application stored in 11.

The orders are followed by internal channel order which provides the change from the current channel to the next channel or to channel 53.

Figure 3 shows the inside memory 20 of a decoder 2 according to the embodiment of the invention more 25 particularly described here, corresponding to the Open TV environment in TV broadcasting.

However, the invention shall of course not be considered as limited to such environment.

The decoder's memory 20 includes within its system a recognition program 21 in order to identify Surfer application from normal applications or other applications such as an Electronic Program Guide (EPG) application.

As already mentioned in the embodiment more particularly described here in a DVB environment, a Surfer is an application presenting all services, and for instance but not limitatively their current and following events of a particular bouquet.

The Surfer is capable of running on top of any TV service and allows the viewer to tune from one channel to another within a bouquet.

During such « channel surfing », the broadcasted Surfer remains alive and active.

For identifying and then retrieving a Surfer from the broadcasted signals, the Surfers are identified and transmitted in a particular service, through the Bouquet Association Table (BAT).

Such service includes a component called „SURF“ identified by a specific flag.

According to the invention, the memory 20 comprises at least a Surfer Cache 22 which is dedicated for storing retrieved Surfers into the decoder. Once a Surfer is stored in decoder's memory, it will be re-launched from memory each time any other application is stopped.

As mentioned the other applications are mainly Electronic Program Guide (EPG) applications that are

being to transmitted in certain services, and normal applications.

An EPG is structurally identical to a normal application and displays information on the contents 5 of events, i.e. TV, Radio or Data events.

It is reminded that an event is information data on a service, is only related to the content and for instance is not related to any frequency or transponder.

10 It is also reminded that a service consists of video, audio or application data, or any combination of these three components.

A plurality of video, audio or application data streams may generally occur within one service.

15 All other broadcasted applications different from Surfer are defined as normal applications or EPG.

They are transmitted as default applications component in any service. They are not privileged by authorisation flags, for instance for changing 20 channel or for getting stored in the memory of the decoder.

The memory 20 further includes a built-in Banner 23 as already described in reference to figure 2, and additional registers of memory 24, for storing and 25 processing a normal or EPG application.

The interactive decoder according to the invention having knowledge about normal and Surfer applications, normal applications are directly loaded (arrows 26) and executed into decoder's memory 24.

On the contrary a Surfer application (arrow 27) is first stored into the Surfer Cache 22 and then executed from its Cache into the memory 24.

Once a Surfer is loaded in its Cache and running,
5 the decoder is under control of this Surfer.

Referring to the flowchart 28 of the lower part of figure 3, any navigation event, like channel up (Key P+ or change of channel to another channel (key 4 for instance), is first going through the
10 recognition process (diamond 29 and 30) to find out if they need to get forwarded to the built-in Banner in 23 or to the downloaded Surfer in 22, the keys pressed may then having another result.

In case the decoder is Surfer controlled (arrows
15 31, 32) the navigation events are rerouted to the Surfer application and the built-in Banner is disabled.

The look and feel of the request channel is therefore different from the built-in look and feel.

20 Figure 4 is a block diagram highlighting the selection layer as provided in the embodiment of the invention more particularly described here, showing that a service is selected either using a Banner or a Surfer. If a Surfer is detected (diamond (33), the
25 system tune to the service with Surfer (step 34), load in the Cache (step 35), and start the Surfer (step 36) for running service (step 37). If not, the Banner is selected in 38, the service list is

displayed in 39, and the service is selected (step 40) for service running (step 37).

In an advantageous embodiment the Surfer which is running may be in a transparent mode by default. For 5 instance, the Surfer may only process specific features (i.e. displaying an additional graphic) while the viewer is watching TV.

In such case, a user can make a Surfer going from transparent mode, in a reversible fashion.

10 The Surfer which is temporarily stored in the decoder RAM-memory, allows a fast access whenever it is requested.

For this it is designed and implemented to fit in the decoder's dedicated Cache memory space (Surfer 15 Cache) which may be, for instance of 64 to 128 kbites.

Therefore storing in the Cache or « caching » the Surfer allows any bouquet operator to add channels of other transponders to his bouquet, which does not 20 contain any Surfer application.

In case the Surfer is requested to quit, it can be reloaded from memory although there is not anymore connection to any transponder carrying bouquets Surfer, which is one of the results of the invention.

25 Generally there is only one application using the application memory at a time. This requires a reloading process of the Surfer from its Cache after the Surfer has to be stopped when a normal application is processed.

When the displaying of normal applications are finished, the Surfer is re-launched from its Cache.

Figure 5 shows that several Surfers (Surfer 1, Surfer 2, ...) may be available in the environment.

5 The decoder system 41 is then advantageously in charge of presenting an interface 42 using a list of services that allows a viewer to select one particular Surfer.

The system will then reserve a key on the remote 10 control device that leads the viewer back to the Surfer selection layer.

It will now be described more precisely an embodiment of the process of the invention, involving a built-in Banner software, hereafter referred to as 15 the Navigator, and wherein there are several bouquets, i.e. services lists broadcasted within several DVB Bouquet Association Tables (BAT).

The user or viewer uses a remote control device having a keyboard, for instance according to the 20 following table.

List of keys

m = mandatory and o = optional

button		Description
Navigator or surfer selection	m	It is used to launch or activate the Navigator
EPG or Surfer	m	It is used to launch a Surfer or an EPG application
OK	m	It is used to confirm a selection and to activate a process
Arrow Up	m	It is used for user interface
Arrow Down	m	navigation. Any application can
Arrow left	m	use it for its own purpose
Arrow Right	m	
Bookmark	m	It is designed to select bookmark-related features in the EPG or Surfer. It can furthermore be used by any application to mark or unmark a selection or for any other purposes.
Channel Up	o	If it appears on a remote control it shall switch channels only within the currently active service list.
Channel Down		
Colour Red	m	Colours buttons can be used by
Colour Green	m	any application for its own
Colour Yellow	m	purpose
Colour Blue	m	
Numeric Keys		Numeric Keys (labelled from 0 to 9) can be used by any application for its own purposes. Furthermore they can be used by Navigator or Surfer to enter directly any channel/service number in the current service list.
Exit	o	This key is used to finish any application. It is considered as „emergency halt“ whenever the user does not want to continue an application.

Table 1 - Remote Control Buttons

Whenever the viewer powers on the decoder, the Navigator is started. It will then display the Navigator first page or recover the last session before the decoder was turned off. The decoder may 5 also be put on stand by in a way known per se.

A bouquet is then selected from the built-in Navigator.

After selecting a bouquet, the Navigator launches the bouquet's Surfer service, which is advantageously 10 also the bouquet's default service, and finds the Bouquet's Surfer application indicated in the bouquets default service PMT (Program Map Table).

PMT is defined in the MPEG-2 standard [3].

The Surfer application is then downloaded into the 15 decoder and temporarily stored in the Surfer Cache of the decoder's RAM. This portion of memory is designed to hold a bouquet Surfer as long as the viewer stays in the bouquet. After downloading the Surfer, the Navigator will start the Surfer application.

20 Following descriptors in BAT are for instance available :

Linkage Type (LT)	Meaning
1	bouquet carries a <u>Surfer</u> service ID in linkage descriptor is Surfer service (Surfer service == default + surf-track)
2	there is an <u>EPG-Service</u> in bouquet (no Surfer) service ID in linkage descriptor is EPG-service
5	a <u>bouquet-default-service</u> exists service ID in linkage descriptor is default service

Table - Linkage Descriptors in BAT

This provides several type of linkage, i.e. for example :

Linkage Type 1

A linkage descriptor of linkage type 1 tells the
5 Navigator that the bouquet associated with the BAT
contains a broadcasted Surfer application. In order
to load and launch the Surfer first, the Navigator
must tune to the Surfer service. This service is
indicated in the linkage descriptors service ID. The
10 Navigator then will load the application that comes
on the component called « SURF ».

Linkage Type 2

A linkage descriptor of linkage type 2 tells the
Navigator, that the bouquet that is associated with
15 the BAT does not contain a broadcasted Surfer
application but a dedicated EPG application. The EPG
button on the remote control is triggering the
Navigator to tune to the dedicated EPG service and to
launch the application that comes on the default
20 track.

Note : Since no Surfer is presenting the service
list of this bouquet to the viewer, the manufacturers
Navigator of the Built-in Banner is used to present
the bouquets services on the TV screen.

Linkage Type 5

A linkage descriptor of linkage type 5 tells the
Navigator that the bouquet does not contain any EPG
or Surfer, but that one service is representing the
bouquets default service. This service could be used

by the Navigator as a first service to launch, in case the Navigator requires running video/audio in the background on bouquet entry. (this is only true, if Navigator's OnScreen display does not cover the whole screen and that some video needs to be playing in the background).

Following the explanations above, one can identify five meaningful combinations of linkage descriptors which are summarised in Table 3 - linkage type matrix.

		LT1	LT2	LT5
Navigator controlled	I	x		
Navigator controlled	II		x	
	III			x
	IV		x	x
	V			

Table 3 - linkage type matrix

Case I : only linkage type 1 is indicated in the BAT. Using the service ID in the linkage service is also the default TV service. The Navigator will therefore turn on Video and Audio and then starts loading and launching the bouquets Surfer. The Surfer service is implicitly also the default service.

Case II : only linkage type 2 is set in the BAT. The Navigator has to present the bouquets service list and offer surfing capabilities. As soon as the viewer presses the EPG button on the remote control, the navigator will tune to the service that is

indicated in the linkage descriptors service ID and launch the default application.

Case III : only the default service is indicated in the linkage descriptor of linkage type 5 in the 5 BAT. If Navigator's user interface is covering the whole TV screen - no video is running in the background and therefore no video has to be turned on. But if the Navigator is transparent or only partially covering the TV screen then the Navigator 10 shall first tune to the bouquets default service.

Case IV : Linkage type 2 and linkage type 5 are both indicated in the BAT. This means that the Navigator first shall make use of the default service and whenever the user is about to press the EPG 15 button the Navigator is launching the EPG application from the indicated service.

Case V : No linkage descriptor is indicated in the BAT. The Navigator must present the service list coming from the BAT to the viewer. Furthermore the 20 first service in the service list is the default service. No EPG or Surfer is available. The EPG button will release a message explaining that no EPG or Surfer is available for this bouquet.

As mentioned, in the DVB environment, see also 25 reference [1], a Surfer is supporting the understanding of bouquets and are signalled in bouquet association tables - BAT.

A "private data specifier" in the first loop of the BAT indicates that further descriptors have a

certain meaning, a following linkage descriptor type == 1 containing a service ID, of the service that carries the Surfer application.

This service has a track tag "SURF", indicating 5 that this component is carrying the Surfer application.

For Surfers indicated in the DVB service type, an example of routine to build up decoder-internal knowledge of available Surfers and Bouquet 10 Association Tables is given hereafter :

```
anzServices = Number of all available Services;
for n=1 to anzServices
    if (Service_type of Service n) .eq. Surfertype
        then store Surfer details in decoder's Surfer list
15    next n
end for
```

For Surfers indicated in a DVB Bouquet Association Tables, an example of the following routine applies 20 to build up internal knowledge of available Surfers:

```
nBAT = Number of available bouquet association tables
for n=1 to nBAT
    if      BouquetAssociationTable[nBAT]      contains
25    Surfer_type indicator
        then store Surfer details in decoder's Surfer list
    next nBAT
    end for
    Scanning for DVB bouquet association tables (BAT)
30 or service description tables (SDT) is processed
according to DVB SI standard, the disclosure of which
is incorporated herein by reference.
```

More specifically OpenTV applications
are using flags to indicate

application properties. The OpenTV inflash flag for applications and for each application module will here tell the recognition process in the system that a particular application is a Surfer application.

5 The application flag is also used to avoid unexpected loading of Surfer applications.

Diagrams of figures 6 and 7 illustrate the implementation of a Surfer according to the embodiment of the invention more particularly
10 described here.

In the diagrams, rectangles represent states, rectangles with right and left borders refer to other programs (not developed particularly), diamonds are yes/no test and diamonds containing circles refer to
15 Remote Control button press/release events.

Once a Surfer is downloaded, it can be started in
50.

A Surfer can start in transparent or in visible mode. It is up to the Surfer to find out if it needs
20 to display anything on screen or if it is better to stay in transparent mode.

Each time the Surfer is launched it will explore current service environment in order to guarantee proper service handling.

25 Since in this embodiment of the invention, Surfer is a bouquets control-application, it is launched on bouquet entry and also on bouquet exit. Launching the Surfer on bouquet-exit provides a way of cleaning up the current bouquet session.

The Surfer will then identify events in its message queue according to its status while launching.

Diagram of figure 6 illustrates, as a non 5 limitative example, a Surfer behaviour model.

Other Surfers might offer less or more functionalities but the Navigator needs to be implemented in order to interoperate with such described Surfer behaviour.

10 A test 52 to quit the Message in queue is provided either to clean up in 53 and exit, or to restore the Service and test if the Surfer is still connected in 54.

15 If the Surfer is connected, the Surfer runs in visible mode (step 56).

Then, pressing a Key among [EPG] [0], [1], ..., [9] [P+] [P-] [OK] allows for selecting services (step 58) or for changing service (test 60) and/or exiting the visible mode.

20 If the Surfer is connected, a test 55 on the Surfer Key in queue is provided, a display current service (test 57) is performed, and if it is a pure data service, a wait screen (step 59) is displayed before going to step 56 (i.e. running in visible mode 25 and activating or not the EPG press button).

If the test 55 (Surfer Key in queue) is negative, a test 61 (p+/p- Key in queue) is undertaken for either making the Surfer visible (step 62), perform the program select service (step 64) which will be

described hereafter, and then eventually return to Transparent mode (67) or restore service environment (step 63), for instance if no connection at all tune to default service.

5 Then a test 65 « current service = pure data service » is performed.

Either a service is selected (step 66) and it is returned to step 56 for Surfer running in visible mode, or the Surfer is running in Transparent mode
10 (step 67).

Selected Key [EPG], [OK], [P+], [P-] are then accessible either for making Surfer visible (step 68, from [EPG] or [OK] Keys) or for keeping it transparent ([P+], [P-] Keys).

15 Figure 7 provides an example of the re-launching of a Surfer according to an embodiment of the invention.

In any Surfer controlled bouquet, decoder's Navigator has to re-launch the Surfer each time any
20 broadcasted application dies. The Navigator then controls the Remote Control (RC) keys as indicated on figure 7 and proceeds with appropriate actions, for instance requests any running application to quit and re-launch Surfer from memory.

25 When a service is running (step 70), the viewer may use any of RC keys, i.e. Exit Key, p+/p- Key, Surfer Key, Surfer selection Key, Bookmark Key and performs a test 72 for checking if a Cached Surfer is available.

If the viewer wishes to exit the application in 71, then a test 73 on Surfer Running is performed.

If the Surfer is not running, a test 75 on the availability of the Surfer is undertaken, and if the 5 response is no (line 79), the application is shut in 81.

If a Cached Surfer is available (line 83), a test 85 on the fact that an application is or not running is performed. If it is the case, the application is 10 shut down in 87, if not, the Surfer is started in 89.

If the test 73 on the Surfer is positive, then the Surfer is shut in 72, and it is returned to the Navigator 74.

An application crash test 76 can be provided, 15 leading to a test 78 on the service. If the current service is pure data service, a step 80 avoiding black TV screen is performed before coming back to application in 82.

If p+/p- Key is pressed, a test to determine if 20 Cached Surfer is available is performed in 84.

If the response is positive, a step 86 starting the Surfer is provided, and Surfer is taking of request for service change.

If no Cached Surfer is available, a test 88 to 25 find out if an application is running is undertaken. If it is the case, the application is shut down in 92, if not a service is selected in 92 with p+/p- Key.

When the Surfer Key is pressed, a test 91 on the availability of a Cached Surfer is provided.

If it is the case, the Surfer is started in 93 by pressing the EPG Key.

5 If not, a test on the Bouquet Service List is made in 95, and in case of positive response a test 97 on Surfer linkage is provided, which allows to select services (step 99) or goes to a test 101, also related to the Bouquet Service list test 95, for
10 tracking the « SURF » features in 101.

If no « SURF » track is found, no Surfer is displayed. If a « SURF » track is found, a test 103 on inflash Flag = Surfer Application is performed.

If not, a normal application is started in 105.

15 Otherwise, the Surfer is loaded in 107 and the Surfer is started in 109.

If the Surfer Selection Key is pressed, a test 94 on Cache Surfer availability is provided.

20 If the response is positive, a test 96 on the running of a Surfer is performed and if it is the case, the Surfer is shut down in 98.

If it is a negative response on the running, then a test 100 on application running is provided. If the response is positive the application is shut in 102.

25 If the response is no, or after such stopping of the application, the Surfer is started in 104, to be shut in 98.

Then the Surfer selection layer 106 is reached.

The other path for reaching such layer goes through the negative response to test 94.

The test 108 « Application running » is then provided.

5 If not, or if yes after shut down of application in 110, the Surfer Selection layer is finally also reached.

Depending on the type of service one is currently selecting, Navigator or Surfers need to perform
10 similar actions. For instance the program « Select Service » in case of a data Service that carries an Open TV application, including video and audio components, should not get opened while any application is being downloadable into decoders
15 memory and a wait-screen shall be displayed. Furthermore, any service containing Open TV components has a default Open TV component. Usually this Open TV component is a broadcasted application.
In some services EPG\0 or SURF component is
20 considered being service's default component or track.

The specific program « Select Service » (not represented by a diagram here) then leads back to the process that is performing Channel changing. For
25 instance a Surfer will keep control after changing services and staying visible in OSD, while Navigator might launch an application component after changing Services right away.

In fact, the system remains in Surfer-control-state as long as the viewer is not requesting the built-in control level (the Banner). On this level the viewer can select other services or choose any 5 other Surfer that is available in networks.

The IRD system must guarantee that there is only one Surfer at a time and running, if the IRD is in Surfer-control-state. In system environments, where several applications can run at a time, a Surfer 10 application is considered as a privileged application with a certain fixed set of system resources. The Surfer is not going to be edged out as long as the viewer does not request the decoder to return into the service selection layer.

15 In system environments, where multiple concurrent applications are being managed by the system, Surfers are only privileged applications.

In system environments, where only one application is allowed being executed at a time, Surfer must be 20 re-launched after any ordinary application has been temporary launched and finished execution, as described hereabove.

In Surfer-control state, Surfer shall be loaded and temporary stored into its dedicated part of IRD's 25 memory or Cache, until the viewer returns to the built-in control state. Also, in Surfer-control state, the Surfer must be accessible in any case, since the Surfer is in charge of maintaining the System environment. Again, this allows the viewer to

tune to other networks and re-launching the Surfer, even though there is temporary no physical connection to the network to which the Surfer was previously loaded from.

5 Caching the Surfer into the IRD's memory will also accelerate Surfer start time on re-launch event.

In the Surfer-controlled state, the decoder is either running the current Surfer or an ordinary application. As soon as an ordinary application is 10 finished the Surfer will come up again. After the viewer requested the built-in state, the Surfer will need to run a clean up process in order to make a clean sweep. The IRD does not consider any Surfer in the built-in control-state for any navigation 15 purposes. It is therefore unimportant if the Surfer application is physically erased. The decoder system will maintain the knowledge of being in either state.

In one embodiment, in order to guarantee that only one Surfer is taking control of the decoder's 20 navigation functions, decoder's Surfer resources are unique resources in the decoder system. A Surfer task or a Surfer memory slot exists only once. Moreover logical flags indicating the IRD's status are present in the decoders system only once. There is no 25 knowledge of competing Surfers in the system. It is therefore at any time clear what part of the system is handling service navigation events.

Apart from links to particular services the IRD built-in software will allow the viewer to select a

tune to other networks and re-launching the Surfer, even though there is temporary no physical connection to the network to which the Surfer was previously loaded from.

5 Caching the Surfer into the IRD's memory will also accelerate Surfer start time on re-launch event.

In the Surfer-controlled state, the decoder is either running the current Surfer or an ordinary application. As soon as an ordinary application is
10 finished the Surfer will come up again. After the viewer requested the built-in state, the Surfer will need to run a clean up process in order to make a clean sweep. The IRD does not consider any Surfer in the built-in control-state for any navigation
15 purposes. It is therefore unimportant if the Surfer application is physically erased. The decoder system will maintain the knowledge of being in either state.

In one embodiment, in order to guarantee that only one Surfer is taking control of the decoder's
20 navigation functions, decoder's Surfer resources are unique resources in the decoder system. A Surfer task or a Surfer memory slot exists only once. Moreover logical flags indicating the IRD's status are present in the decoders system only once. There is no
25 knowledge of competing Surfers in the system. It is therefore at any time clear what part of the system is handling service navigation events.

Apart from links to particular services the IRD built-in software will allow the viewer to select a

Surfer from a list of Surfers. The IRD therefore need knowledge about existence and location of any Surfer applications.

The way of advertising and indicating Surfer application is network specific and does not interfere with the service browser concept. Surfers could be indicated for instance as a particular service type, or a specific component tag within a channel or program. It is then the decoder's responsibility to find Surfers across the networks and to present links to those browser applications.

Additional remarks are made concerning the decoder. In the prior art any application is allowed to take advantage of decoders' resources. However some resources are limited and sensitive in terms of the proper behaviour and performance of any decoder. The decoders' permanent memory in EEPROM or Flash is one of the most critical points.

With the embodiment of the invention more
particularly described here, it will now be defined
the following:

File System Policy

Any application could allocate all remaining space in decoders File System for a very long time. Since there is no System administrator any other application has to wait for the expiration date of this file before it can make use of the decoders file system. In order to avoid one broadcaster or service provider blocking the decoder's File System memory to

Surfer from a list of Surfers. The IRD therefore need knowledge about existence and location of any Surfer applications.

The way of advertising and indicating Surfer application is network specific and does not interfere with the service browser concept. Surfers could be indicated for instance as a particular service type, or a specific component tag within a channel or program. It is then the decoder's responsibility to find Surfers across the networks and to present links to those browser applications.

Additional remarks are made concerning the decoder. In the prior art any application is allowed to take advantage of decoders' resources. However some resources are limited and sensitive in terms of the proper behaviour and performance of any decoder. The decoders' permanent memory in EEPROM or Flash is one of the most critical points.

With the embodiment of the invention more particularly described here, it will now be defined the following :

File System Policy

Any application could allocate all remaining space in decoders File System for a very long time. Since there is no System administrator any other application has to wait for the expiration date of this file before it can make use of the decoders file system. In order to avoid one broadcaster or service provider blocking the decoder's File System memory to

other applications, a producer based partitioning of the whole memory is recommended. Any broadcaster or service operator of the environment system of the invention, shall use a range of producer identifications. Only some producers ids are allowed to allocate memory in the File System.

The consumer may select this producer Ids. Actually he is performing the partitioning of his decoders' memory and therefore the consumer decides which broadcaster may take advantage of the decoders resources.

Furthermore any service provider (producer) is in charge of „his“ part of memory. This will lead to a careful use of memory allocation in File System. There is a minimum slot size a consumer might allocate for each broadcaster.

Bouquets' slot size

In a embodiment of the invention, a minimum Slot Size of 32 kbytes shall be available in the decoders permanent memory for any bouquet operator that has been authorised by the consumer to allocate memory in the permanent File-System.

Bouquet Memory Identification

Any application will be identified by a producer ID and an application ID. The producer ID has to be a unique value.

An open TV producer ID contains for example a DVB Bouquet ID in order to identify the Bouquet provider.

This value is used to identify the path in the file system and application is allowed to store data.

Slot management Policy

The decoder shall have 128 Kbytes available for

5 Ram File System and 256 Kbytes of Flash memory available for Flash File System.

Quota Management Policy applies for permanent part of the decoders File System, the Flash File System.

Only application with appropriate authorisation 10 flags are allowed to allocate space in the File System, Flash or Ram. Only applications with appropriate Producer ID and authorisation flags are allowed to use Flash File System. Manufacturer needs therefore to customise the file System usage.

15 Therefore, the decoder needs to maintain a list of valid ProducerIDs, that are allowed to write into the Flash File System. Attached to this list there is a total value, which is currently allocated in each 32Kbytes Slot in Flash File System. Any broadcasted 20 application comes with a figure of how much memory it intends to allocate. In case this number exceeds the remaining space, it could be overwritten and lowered to the maximum left amount. Any further memory allocation is then handled by Open TV and allocation 25 requests that exceeds this limit will be denied.

It will now be described an example of functioning of a system according to an embodiment of the invention.

From a Remote Control Key boards the viewer is selecting a list of Bouquets, i.e. five bouquets among which two include Surfers (Bouquet 1 and 2).

Then one bouquet including a Surfer is selected by 5 pressing the **OK** button, for instance Bouquet 1.

The decoder tunes to the default service of Bouquet 1 and download the Surfer in the Surfer Cache.

10 The Surfer is then launched from the Surfer Cache on the TV screen while the decoder is simultaneously tuned to Bouquet 1.

Then the Surfer is displayed on part of the TV Screen, for instance on top of the running default audio-video service (the default channel of the 15 Bouquet) in the form of a scrolling list mentionning all the channels of the Bouquet.

Then a channel is selected on the list by pressing up and down buttons.

When the right channel is reached, the viewer 20 tunes the decoder to said selected channel by pressing the OK button.

If a Bouquet deprived of a Surfer having a specific look and feel is selected, the viewer has access to normal application including EPG only 25 through the Built-in Banner look and feel.

In case of Surfer, instead of tuning to a regular TV channel (normal application), it is possible to tune to an EPG application which will provide events information on an interactive basis.

Each time, another channel with an application (i.e. not only an audio or video service, but also an EPG application) is selected from the Surfer, the Surfer has to be relaunched from the Surfer Cache.

5 Additional advantages and modifications will readily occur to those skilled in the art. Therefore the present invention in its broader aspects is not limited to the specific details, representative device and diagrams, and illustrated examples shown
10 and described herein.